BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/817,259	03/27/2001	Koji Muramatsu	04329.2552	7688

22852 7590 06/02/2004

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP
1300 I STREET, NW
WASHINGTON, DC 20005

EXAMINER	
NGUYEN, VAN H	
ART UNIT	**PAPER NUMBER**
2126	

DATE MAILED: 06/02/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/817,259	MURAMATSU, KOJI	
	Examiner	Art Unit	
	VAN H NGUYEN	2126	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *27 March 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-9* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-9* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *3/27/01*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-9 are presented for examination.

Claim Objections

2. Claims 5 and 8 are objected to because of the following informalities:

(i) the phrase "*a an* instruction" (claim 5, line 3) should read " *an* instruction"

(ii) the phrase "*a an* instruction" (claim 8, lines 2-3) should read " *an* instruction"

Appropriate correction is required.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 1-3, 5, and 7-9 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

A. The following phrases lack antecedent basis:

(i) the other instruction (claim 1, lines 32-33; claim 2, line 22; claim 3, line 16)

(ii) the received thread identifier (claim 3, line 16)

(iii) the lock (claim 5, line 16; claim 7, line 8; claim 8, line 9; and claim 9, line 10)

(iv) the instruction (claim 7, lines 8-9)

(v) the event processing thread (claim 9, lines 11-12)

(vi) the restart (claim 9, line 13)

B. The following phrases are indefinite:

(i) a client machine (claim 1, line 17)

(ii) a restart request (claim 9, line 15)

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which the subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made

6. Claims 3 and 6-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Lim et al.** (U.S. 6,212,573).

7. As to claim 3, Lim teaches *(abstract)* the invention substantially as claimed including a computer program product that records a program for operating a client machine *(e.g., the client)* to which a server machine *(e.g., the server)* entrusts an instruction in an automatic distributed

processing system *(e.g., a distributed client/server-based object oriented operating system; abstract)*, the program comprising:

- computer readable program code means for making the client machine implement a thread creation function *(e.g., a response record data structure... is created for each request transmitted from a client which requires a reply from a server; col.14, lines 11-23 and fig.9)* of creating a thread *(e.g., the thread; col.14, lines 21-23)*, a thread identifier *(e.g., Thread ID 904 is a unique identifier which corresponds to the thread; col.14, lines 11-23)*; and

- computer readable program code means for making the client machine implement a function of sending another instruction *(e.g., a request identifier (ID) 902...further include "extra" information 908; col.14, lines 11-16)* which is generated while the thread created by the thread creation function processes the instruction to the server machine while appending the received thread identifier to the other instruction *(col.14, lines 24- 42)*.

Lim does teach a client thread which is identified by the thread identifier (*The thread identifier, which identifies the client thread; col.13, lines 16-27)*, but not explicitly teach "a thread that processes an instruction received from the server machine together with a thread identifier, on the basis of the thread identifier."

Lim, however, discloses *"a listen event is sent to the client thread which is identified by the thread identifier (ID) ...the current thread sends a listen event to the client thread which is appropriate for receiving the reply. The thread identifier, which identifies the client thread that transmitted the request corresponding to the request record... After a listen event is sent to an appropriate client thread, the steps which must be executed by the current client thread"* (col.13, lines 16-27).

It would have been obvious to one of ordinary skill in the art to have applied the teaching of Lim for "a thread that processes an instruction received from the server machine together with a thread identifier, on the basis of the thread identifier" in order to reduce the amount of computing overhead and, more importantly, the latency associated with processing requests, thereby improving the overall efficiency of a distributed object computing system.

8. As to claim 8, Lim teaches the invention substantially as claimed including a computer program product that records a program for operating a server machine *(e.g., the server program; col.6, lines 4-10)* that entrusts an instruction in an automatic distributed processing system *(e.g., a distributed client/server-based object oriented operating system; abstract)*, the program comprising:

- computer readable program code means for, when a first instruction is generated during processing of an application after an exclusive lock *(e.g., the operation of a worker thread which services a server side end point... the mutex select lock is acquired by a worker thread... The mutex select lock is a lock which provides exclusive access to the end point for the worker thread which possesses the lock; col.8, lines 10-20 and fig.5),* making the server machine release the lock in correspondence with contents of the instruction, and send the first instruction to an instruction processing thread of the client machine *(col.13, lines 16-27);*

- computer readable program code means for making the server machine end the first instruction upon receiving an end reply of the instruction process in the instruction processing thread *(col.8, lines 30-34);* and

- computer readable program code means for making the server machine process a second instruction sent from an event processing thread of the client machine *(fig.5 and associated text).*

Lim does teach releasing the lock by the server machine *(the mutex select lock is released; col.8, lines 16-27)*, but not explicitly teach "the server machine release the lock in correspondence with contents of the instruction."

Lim, however, discloses *"the mutex select lock is free to be acquired in step 504 after the reading request is processed" (col.13, lines 16-27).*

It would have been obvious to one of ordinary skill in the art to have applied the teaching of Lim for "the server machine release the lock in correspondence with contents of the instruction" in order to allow another worker thread to acquire the lock to perform its activity. Therefore, reducing the amount of computing overhead and, more importantly, the latency associated with processing requests.

9. As to claim 9, Lim teaches the invention substantially as claimed including computer a program product that records a program for operating a client machine to which a server machine entrusts an instruction in an automatic distributed processing system in which the server and client machines are connected via a network *(abstract and col.6, lines 12-36)*, the program comprising:

- computer readable program code means for, when a first instruction is received from the server machine *(col.11, lines 36-40),* making the client machine acquire an exclusive lock and process the first instruction *(e.g., the current client thread acquires the write lock...the client thread writes the request on the connection; col.11, lines 50-55)*, release the lock *(e.g., the write*

lock for the connection is released; col.11, lines 55-57), wait until a restart request is received from the event processing thread *(col.13, lines 40-52),* entrust the end of the instruction to the server machine, and send a restart request to the client machine which is in the wait state *(col.13, lines 40-50);* and

- computer readable program code means for, when a second instruction is generated during a self event process after an exclusive lock, making the client machine entrust the second instruction to the server machine *(col.13, lines 53-65).*

Lim does teach the lock is released by the client *(e.g., the write lock for the connection is released by the current client thread; col.11, lines 55-57)*, but not explicitly teach "release the lock upon completion of the instruction process after the restart."

Lim, however, discloses *"after the request is written on the connection, the write lock for the connection is released by the current client thread" (col.11, lines 50-55).*

It would have been obvious to one of ordinary skill in the art to have applied the teaching of Lim for "release the lock upon completion of the instruction process after the restart" in order to allow another worker thread to acquire the lock to perform its activity. Therefore, reducing the amount of computing overhead and, more importantly, the latency associated with processing requests.

10. As to claim 6, the rejection of claim 9 above is incorporated herein in full. Additionally, Lim teaches computer readable program code means for making the client machine implement a retry request function of sending a retry request to the server machine when the checking function determines that the lock cannot be acquired *(col.10, lines 45-62).*

11. As to claim 7, it is a combination of claim 8 and 9 above. Therefore, note the discussion of claims 8 and 9 above for rejection.

12. Claims 1-2 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Lim et al.** in view of **Nally et al.** (U.S. 6,298,478).

13. As to claim 2, Lim teaches the invention substantially as claimed including a computer program product that records a program *(e.g., the server program, col.6, lines 4-11)* for operating a server machine *(e.g., the server, col.6, lines 14-18)* that entrusts a process *(e.g., a server process, col.6, lines 4-11)* of an instruction to a client machine *(e.g., the client, col.6, lines 14-18)* in an automatic distributed processing system *(e.g., a distributed client/server-based object oriented operating system; abstract),* the program comprising:

- computer readable program code means for making the server machine implement an instruction relay function of appending a thread identifier to an instruction generated during processing of an application of the server machine *(col.12, lines 34-36)* and sending the instruction to the client machine *(col.6, lines 12-18 and col.13, lines 15-27);*

- computer readable program code means for making the server machine implement an instruction distribution function of distributing another instruction which is generated upon an instruction process of the client machine and is appended with the thread identifier to a thread as an entrust source *(col.13, lines 15-27 and col. 14, lines 11-23)*; and

- computer readable program code means for making the server machine implement a function of returning a processing result of the other instruction which is distributed to and processed by the thread of the server machine to the client machine *(col.6, lines 28-36 and*

col.14, lines 40-42).

Lim does teach thread identifiers and threads *(e.g., information which identifies the client thread; col.12, lines 34-36/Thread ID 904 is a unique identifier which corresponds to the thread; col.14, lines 11-23),* but is silent on looking up a table that manages a relationship between thread identifiers and threads.

Nally discloses looking up a table that manages a relationship between thread identifiers and threads *(e.g., a lookup table could be used, where the table index is the identifier of the current thread, and the table value is the transaction associated with that thread; col.16, lines 34-44).*

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Nally and Lim because Nally's teaching would have provided the capability for eliminating the requirement of keeping a pointer in the threads.

14. As to claim 1, the rejection of claim 2 above is incorporated herein in full. Additionally, Lim further teaches:

- a client instruction distribution thread for receiving the instruction sent from the instruction relay thread of the server machine together with the thread identifier, creating a thread that processes the instruction, and passing the received instruction to the created thread together with the thread identifier *(col.6, lines 55-60 and col.13, lines 15-27);* and

- an instruction processing thread for processing the received instruction in collaboration with a higher-level library of the client machine, and for, when another instruction is generated upon processing the received instruction or the processing of the received instruction is complete, sending the other instruction or a processing end reply appended with the thread

identifier to the instruction distribution thread of the server machine *(col.13, lines 23-39).*

15. Claims 4-5 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Lim et al.** in view of **Snyder et al.** (U.S. 6,640,255).

16. As to claim 4, Lim teaches the invention substantially as claimed including an automatic distributed processing system in which a server machine and client machine are connected via a network *(abstract and col.6, lines 12-36)*, wherein each of the server and client machines comprises:

- an instruction relay thread for, when an instruction is generated upon processing of a self application after an exclusive lock, acquiring a lock and relaying the instruction to a partner machine *(col.8, lines 8-34 and col.38-58),* and an instruction processing thread for receiving and processing the instruction from the instruction relay thread *(col.13, lines 16-27),*

- at least the instruction processing thread of the client machine comprises means for receiving the instruction from the server machine *(col.13, lines 16-27)*, checking if a self machine can acquire a lock, and sending a retry request to the server machine if the lock cannot be acquired *(col.10, lines 45-62)*, and means for acquiring the lock if the lock can be acquired, and sending a reply upon completion of processing of the instruction, and releasing the lock *(col.11, lines 36-58)*, and

- at least the instruction relay thread of the server machine comprises means for making a retry that temporarily releases the lock, then reacquires the lock, and relays the instruction again upon receiving the retry request from the client machine, and means for releasing the lock upon receiving the reply indicating end of the instruction from the server machine *(col.11, lines 50-*

58).

Lim does not explicitly teach means for making a retry that temporarily releases the lock, then reacquires the lock.

Snyder discloses means for making a retry that temporarily releases the lock, then reacquires the lock *(e.g., a Lock function to explicitly reacquire a mutex after the lock has been released, a Read_Lock function to reacquire a lock for a read operation after the lock has been released, a Write_Lock function to reacquire a lock for a write operation after the lock has been released ; col.13, lines 61-67).*

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Snyder and Lim because Snyder's teaching would have provided the capability for allowing a thread to acquire a lock for performing its activity. Therefore, reducing the amount of computing overhead and, more importantly, the latency associated with processing requests.

17. As to claim 5, the rejection of claim 8 above is incorporated herein in full. Additionally, Lim further teaches setting the server machine setting itself in a reception wait state *(col.8, lines 42-47).*

Lim, however, does not teach reacquiring the lock.

Snyder discloses reacquiring the lock *(e.g., a Lock function to explicitly reacquire a mutex after the lock has been released; col.13, lines 61-67).*

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Snyder and Lim because Snyder's teaching would have provided the capability for allowing a thread to acquire a lock for performing its

activity. Therefore, reducing the amount of computing overhead and, more importantly, the latency associated with processing requests.

Conclusion

18. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- Dwyer III (U.S. 6418442) teaches "Method and apparatus for providing thread-specific computer system parameters"

- Blazo et al. (U.S. 6272518) teaches "System and method for porting a multithreaded program to a job model"

- Brobst et al. (U.S. 6125382) teaches "Distributed thread mechanism and method"

- Fukumoto et al. (U.S. 5892944) teaches "Program execution and operation right management system suitable for single virtual memory scheme"

- Bedy et al. "The design and construction of a use-level kernel for teaching multithreaded programming" 1999 IEEE, pp. 24-29.

- Burns et al. "Semi-preemptible locks for a distributed file system" 2000 IEEE, pp. 397-404.

19. Any inquiry concerning this communication or earlier communications from the examiner should be directed to VAN H NGUYEN whose telephone number is (703) 306-5971. The examiner can normally be reached on Monday-Thursday from 8:30AM - 6:00PM. The examiner can also be reached on alternative Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (703) 305-9678.

The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VHN

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100